Theory of Computing – Examples

v.0.6 - 1/3/2023

Michele Amoretti
Quantum Information Science
University of Parma

QIS.UNIPR.IT
QUANTUM INFORMATION SCIENCE @ UNIPR

Contents

Quantum Information Science
University of Parma
Parco Area delle Scienze
43124 Parma
Italy
http://www.qis.unipr.it

Preface

This is a collection of examples related to the Theory of Computing part of the High Performance Computing course (M.Sc. in Computer Engineering, University of Parma).

1 Information Theory

1.1 Entropy – Horse Racing

Let us consider eight horses with win probabilities $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}$. To communicate the race winner, we may use 3 bits, as to enumerate the eight horses we need actually 3 bits. However, taking into account the fact that the horses have different probabilities of being announced as winners, we may choose a clever binary representation, namely:

$$0, 10, 110, 1110, 111100, 111101, 111110, 111111.$$

In this way, the average description length is 2 bits.

Let's now compute the entropy of the random variable X representing win announcements:

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4}... = 2 \quad \text{bits} \tag{1}$$

We know that entropy is the average amount of information produced by a stochastic source of data. With this example, we have seen that the entropy of a random variable can be also interpreted as the *lower bound* on the average number of bits required to represent the random variable.

1.2 Mutual Information – Binary Symmetric Channel

Let us consider a noisy channel where an input bit b is received as $\bar{b}$ with probability p (as b with probability $1-p$), as illustrated in Fig. 1. This model, denoted as Binary Symmetric Channel, is the most simple model of noisy channel. Let us call X the random variable representing the transmitted bits, and Y the one representing the received bits. Their mutual information is

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - H_b(p) \tag{2}$$

where $H_b(p)$ is the binary entropy. The previous equation can be proved by observing that $\{Y|X = x\}$ is a Bernoulli random variable (when x is transmitted, the value of Y is x with probability $1 - p$ and $\bar{x}$ with probability p).

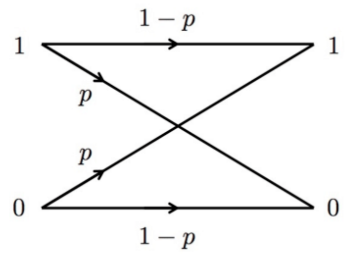

Figure 1: Binary Symmetric Channel model.

The channel capacity is

$$C = \max_{p(x)} I(X;Y) = \left(\max_{p(x)} H(Y)\right) - H_b(p) \tag{3}$$

To compute C, we observe that $H(Y)$ is maximized to 1 when X has uniform distribution ($p(x)$ s.t. $p(0) = p(1) = \frac{1}{2}$), for which also Y has uniform distribution independently of p (the proof is left for exercise). On the other hand, $H_b(p)$ does not depend on $p(x)$. Therefore, the channel capacity turns out to be:

$$C = 1 - H_b(p) = 1 + p \log p + (1-p) \log(1-p) \tag{4}$$

We can observe that, when $p = 0$ or $p = 1$, then $C = 1$. The minimum capacity, $C = 1/2$, arises when $p = 1/2$.

1.3 Kolmogorov Complexity – Find the Random String

Tell if the following strings are Kolmogorov random:

33333333333

31415926535

84354279521

They all have the same probability 10^{-11} of being randomly extracted from the set of 11-digit strings. However, $C(x) < 11$ for the first string (which is $\{3\}^{11}$) and also for the second one (which is $\pi \cdot 10^{10}$). Only the last string is Kolmogorov random, having $C(x) = 11$.

2 Computability

2.1 Mapping Reducibility vs. Turing Reducibility

Let us recall the definition of mapping reducibility:

$$A \leq_m B \Leftrightarrow [w \in A \Leftrightarrow f(w) \in B]. \tag{5}$$

This definition implies that, given w, there is a Turing Machine M^B that computes $f(w)$ and, thanks to the oracle for B, tells whether $f(w) \in B$, i.e., $w \in A$. This means that M^B can decide A, i.e., that $A \leq_T B$. In conclusion, mapping reducibility implies Turing reducibility.

Instead, Turing reducibility does not imply mapping reducibility. Indeed, Turing reducibility means that there is an oracle Turing machine M^B able to decide A. This definition does not imply the existence of $f()$ such that $w \in A \Leftrightarrow f(w) \in B$.

2.2 $\overline{A_{TM}}$ is not mapping reducible to A_{TM}

Let us define the language

$$\overline{A_{TM}} = \{(M, w) | M \text{ is a TM and } M \text{ rejects } w\}. \tag{6}$$

We know that A_{TM} is Turing-recognizable by a Universal Turing machine (UTM). Now we prove that $\overline{A_{TM}}$ is not Turing-recognizable.

If both languages were Turing-recognizable, then A_{TM} would be decidable, i.e., there would exist a TM that halts for all (M, w). Since we know that A_{TM} is not decidable, the initial assumption was wrong.

Now observe that, being $\overline{A_{TM}}$ not Turing-recognizable, there is no $f()$ such that $[(M, w) \in \overline{A_{TM}} \Leftrightarrow f(M, w) \in A_{TM}]$. To have such an $f()$, we would need a Turing machine that recognizes $\overline{A_{TM}}$.

We conclude that $\overline{A_{TM}}$ is not mapping reducible to A_{TM}.

3 Computational Complexity

3.1 GCD – Euclidean Algorithm

Let us consider two positive integers a and b.

1. Find q_0 and r_0 s.t. $a = q_0 b + r_0$.
2. Find q_1 and r_1 s.t. $b = q_1 r_0 + r_1$.
3. Repeatedly solve $r_i = q_{i+2} r_{i+1} + r_{i+2}$ until $r_n = 0$.
4. $GCD(a, b) = r_{n-1}$.

Exercise: find $GCD(125, 75)$ using the Euclidean Algorithm.

3.2 NP-Complete Problems

- **SAT**. Given n Boolean variables $x_1, .., x_n$, is there at least one configuration of the variables s.t. $f(x_1, .., x_n) = 1$?
- **3SAT**. Given n Boolean variables $x_1, .., x_n$ and a set of clauses that each one relates at most 3 variables, is there at least one configuration of the variables s.t. the clauses evaluate to 1?
- **CircuitSAT**. Given n Boolean variables $x_1, .., x_n$ and a set of clauses $x_{i+1} = f_i(x_1, .., x_i)$ with $i \geq n$, is there at least one configuration of the variables s.t. the clauses evaluate to 1?
- **Map Coloring**. Are k colors sufficient to color an arbitrary map so that no two adjacent features have the same color? With $k = 2$, the answer can be found in polynomial time in the number of features (it is sufficient to find a vertex with an odd number of incident edges). With $k \geq 4$, it is always possible to color the map so that no two adjacent features have the same color. With $k = 3$, the problem is NP-Complete.
- **3-Partition**. Given $3n$ numbers, decide whether they can be split into triples of equal sum.
- **Bin Packing**. We have an unlimited number of bins each of capacity B, and n objects of sizes s_1, s_2, etc. s.t. $0 < s_i \leq B$. Given k, is there a packing using no more than k bins?

- **Traveling Salesman Problem (TSP)**. Given a graph and an integer B, is there a cycle through all the vertices such that the total weight of the edges used is at most B?

References

[1] S. Arora, B. Barak, *Computational Complexity – A Modern Approach* Cambridge University Press, 2009.